

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-065858

(43)Date of publication of application : 09.03.1999

(51)Int.Cl.

G06F 9/46

(21)Application number : 09-229895

(71)Applicant : NEC AEROSPACE SYST LTD

(22)Date of filing : 26.08.1997

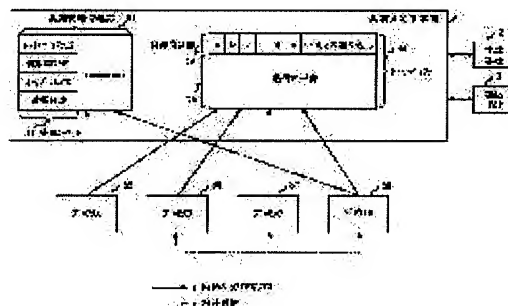
(72)Inventor : KISHI TOSHIYUKI
OKABE KAZUTOSHI

(54) INTER-PROCESS COMMUNICATION SYSTEM USING COMMON MEMORY

(57)Abstract:

PROBLEM TO BE SOLVED: To easily perform an inter-process communication using the common memory at a high speed.

SOLUTION: A common management information part 31 is provided in the common memory space 1 and a communication source process A acquires a usable shuttle buffer ID therefrom and secure a shuttle buffer 32 with its ID. The shuttle buffer 32 has a control information part 33 which holds an owner process ID, etc., and a communication part 34 which holds communication data. A communication destination process B reads the contents of the shuttle buffer out on the basis of the ID and decreases the nonreference process number 33C of the control part of the shuttle buffer after completing reference. The shuttle buffer is released when the reference number 33C of only the control information part reaches 0.



1

【特許請求の範囲】

【請求項 1】 共有メモリ空間を利用したプロセス間通信方式において、通信に使用する前記共有メモリ空間に制御情報部と通信データ部とを付加した共有メモリ領域を備え、前記制御情報部は、前記共有メモリ領域を獲得したプロセスの識別を示すオーナープロセス ID と、前記共有メモリ領域を参照するプロセスの数を示す参照プロセス数と、前記共有メモリ領域を参照中またはまだ参照していないプロセスの数を示す未参照プロセス数と、前記共有メモリ領域を参照するプロセスの識別を示す参照プロセス ID と、前記参照プロセス ID に示されたプロセスが前記共有メモリ領域を参照したのかどうかを示す参照済みフラグとを含み、前記制御情報部を参照し、変更する編集手段と、前記編集手段において、前記未参照プロセス数の値から前記共有メモリ領域を参照するプロセスがなくなったことを検出した場合は前記共有メモリ領域を解放する解放手段とを有することを特徴とする共有メモリを使用したプロセス間通信方式。

【請求項 2】 前記共有メモリ空間に前記共有メモリ領域の使用状況を管理する共通管理情報部を備えたことを特徴とする請求項 1 記載の共有メモリを使用したプロセス間通信方式。

【請求項 3】 前記共通管理情報部は、前記共有メモリ領域の 1 つ毎に対応した管理テーブルを複数持つことを特徴とする請求項 2 記載の共有メモリを使用したプロセス間通信方式。

【請求項 4】 前記編集手段において、前記参照プロセス数を変更し、前記参照プロセス ID と前記参照済みフラグを設定することにより、前記共有メモリ領域に設定された前記通信データ部の内容を他のプロセスに受け渡すことを特徴とする請求項 1, 2, または 3 記載の共有メモリを使用したプロセス間通信方式。

【請求項 5】 共有メモリ空間を利用したプロセス間通信方式において、通信に使用する前記共有メモリ空間に制御情報部と通信データ部とを付加した共有メモリ領域を備え、通信元プロセスは、前記制御情報部にプロセス間の制御のための制御情報を通信データ部に通信するためのデータをそれぞれ設定し、使用している前記共有メモリ領域の場所を第 1 の通信先プロセスに通知する第 1 の通知手段と、前記第 1 の通知手段によって通知された前記第 1 の通信先プロセスは、前記共有メモリ領域を参照し、受信した前記通信データを第 2 のプロセスに通知する必要がある場合は、既に設定された前記制御情報を変更し第 2 の通信先プロセスの ID を前記制御情報部に新たに設定した後、前記第 2 の通信先プロセスに通知する第 2 の通知手段と、第 2 の通知手段によって通知された前記第 2 の通信先プロセスは、前記第 1 の通知手段によって通知された前記第 1 のプロセスと同じ動作を行う動作手段と、前記第 1 の通知手段、前記第 2 の通知手段、または前記動作手段によって通知された全てのプロ

2

セスの前記共有メモリ領域への参照が完了した時点で、前記共有メモリ領域を解放する解放手段とを有することを特徴とする共有メモリを使用したプロセス間通信方式。

【請求項 6】 前記制御情報部は前記共有メモリ領域を獲得したプロセスの識別を示すオーナープロセス ID と、前記共有メモリ領域を参照するプロセスの数を示す参照プロセス数と、前記共有メモリ領域を参照中またはまだ参照していないプロセスの数を示す未参照プロセス数と、前記共有メモリ領域を参照するプロセスの識別を示す参照プロセス ID と、前記参照プロセス ID に示されたプロセスが前記共有メモリ領域を参照したのかどうかを示す参照済みフラグとを含むことを特徴とする請求項 5 記載の共有メモリを使用したプロセス間通信方式。

【請求項 7】 前記共有メモリ空間に前記共有メモリ領域の使用状況を管理する共通管理情報部を備え、前記共通管理情報部は前記共有メモリ領域の 1 つ毎に対応した管理テーブルを複数持つことを特徴とする請求項 5 または 6 記載の共有メモリを使用したプロセス間通信方式。

【請求項 8】 前記管理テーブルは、前記共有メモリ空間での前記共有メモリ領域の場所の識別を示す共有メモリ領域 ID と、前記共有メモリ領域の使用の有無を示す使用中フラグと、前記共有メモリ領域を確保したプロセスの識別を示すオーナープロセス ID と、前記共有メモリ領域を獲得した日時を示す取得日時とを含むことを特徴とする請求項 3 または 7 記載の共有メモリを使用したプロセス間通信方式。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、コンピュータプログラムにおけるプロセス間の通信に関し、特に共有メモリを使用したプロセス間通信に関する。

【0002】

【従来の技術】一般的にプロセス間の通信で大量のデータをやり取りする場合、通信先が同一のコンピュータであるならば共有メモリを使用する方法が最も性能が良い。

【0003】しかし、複数のプロセスに同じデータを受け渡したり、更に、受け取ったプロセスがまた別のプロセスにそのデータを受け渡したりしようとすると、受け渡し用に取得した共有メモリ領域を解放するための制御が複雑になるばかりでなく、そのための通信が頻繁に発生し、システムの性能を低下させる。また、共有メモリの取得・解放を単純にするために、予め取得した共有メモリ領域を使い回してプロセス間通信を行おうとした場合、今度は、受け渡し先のプロセスの参照が終わるまで次の書込が待たされたり、受け渡し先の全てのプロセスの参照が完了したことを通信元のプロセスに知られるための通信が必要になる。

【0004】例えば、特開平 7 - 1 2 9 4 1 6 公報に開

3

示されているように、通信元プロセスが共有メモリの取得、および解放を行っていた。従って、通信先プロセス全てが参照し終わった後に、通信元プロセスに共有メモリの解放要求を通知する必要があった。

【0005】また、特開平 4 - 3 6 4 5 5 0 公報に開示されているように、共有メモリに管理情報を設定し、通信元プロセスが共有メモリの解放を意識せずに済んでいるが、通信先プロセスが通信元プロセスの管理情報を操作するので、複数のプロセスに同時に同じ領域（共有メモリ）を使用してデータを送信することは出来ない。すなわち、1 対多数の通信は考慮されていなかった。

【0006】

【発明が解決しようとする課題】第 1 の問題点は、特開平 7 - 1 2 9 4 1 6 公報のように、通信元プロセスがプロセス間通信で使用済みとなった共有メモリを削除する場合、共有メモリの参照完了を通知するため通信を行う必要があり、プログラム（特に、通信元プロセス）の制御が複雑になるばかりではなく、通信量の増大、並びに通信元プロセスの負荷増大によるシステム性能の低下が発生することである。

【0007】第 2 の問題点は、特開平 4 - 3 6 4 5 5 0 公報のように、通信先プロセスが共有メモリ上の制御情報を操作することによりその領域を未使用状態にする場合、同じ共有メモリ領域を使用して複数のプロセスにデータを受け渡すことができないことである。その理由は、通信先のプロセスのどれか一つが制御情報を操作してしまうとその共有メモリ領域が未使用の状態になり、次の通信に使われてしまうからである。

【0008】本発明の目的は、共有メモリを使用したプロセス間通信を高速かつ容易に行うことにある。

【0009】本発明の他の目的は、1 対多数の通信を可能にすることにある。

【0010】本発明の他の目的は、使用者に意識させない構造とし、プロセス間通信の生産性の向上を高めることにある。

【0011】

【課題を解決するための手段】本発明の共有メモリを使用したプロセス間通信方式は、共有メモリ空間を利用したプロセス間通信方式において、通信に使用する前記共有メモリ空間に制御情報部と通信データ部とを付加した共有メモリ領域を備え、前記制御情報部は、前記共有メモリ領域を獲得したプロセスの識別を示すオーナープロセス ID と、前記共有メモリ領域を参照するプロセスの数を示す参照プロセス数と、前記共有メモリ領域を参照中またはまだ参照していないプロセスの数を示す未参照プロセス数と、前記共有メモリ領域を参照するプロセスの識別を示す参照プロセス ID と、前記参照プロセス ID に示されたプロセスが前記共有メモリ領域を参照したのかどうかを示す参照済みフラグとを含み、前記制御情報部を参照し、変更する編集手段と、前記編集手段におい

4

て、前記未参照プロセス数の値から前記共有メモリ領域を参照するプロセスがなくなったことを検出した場合は前記共有メモリ領域を解放する解放手段とを有することを特徴としている。

【0012】上記の他に、前記共有メモリ空間に前記共有メモリ領域の使用状況を管理する共通管理情報部を備えたことを特徴としている。

【0013】前記共通管理情報部は、前記共有メモリ領域の 1 つ毎に対応した管理テーブルを複数持つことを特徴としている。

【0014】前記編集手段において、前記参照プロセス数を変更し、前記参照プロセス ID と前記参照済みフラグを設定することにより、前記共有メモリ領域に設定された前記通信データ部の内容を他のプロセスに受け渡すことを特徴としている。

【0015】また、本発明の共有メモリを使用したプロセス間通信方式は、共有メモリ空間を利用したプロセス間通信方式において、通信に使用する前記共有メモリ空間に制御情報部と通信データ部とを付加した共有メモリ領域を備え、通信元プロセスは、前記制御情報部にプロセス間の制御のための制御情報を通信データ部に通信するためのデータをそれぞれ設定し、使用している前記共有メモリ領域の場所を第 1 の通信先プロセスに通知する第 1 の通知手段と、前記第 1 の通知手段によって通知された前記第 1 の通信先プロセスは、前記共有メモリ領域を参照し、受信した前記通信データを第 2 のプロセスに通知する必要がある場合は、既に設定された前記制御情報部を変更し第 2 の通信先プロセスの ID を前記制御情報部に新たに設定した後、前記第 2 の通信先プロセスに通知する第 2 の通知手段と、前記第 1 の通知手段、前記第 2 の通知手段、またはによって通知された全てのプロセスの前記共有メモリ領域への参照が完了した時点で、前記共有メモリ領域を解放する解放手段とを有することを特徴としている。

【0016】前記制御情報部は前記共有メモリ領域を獲得したプロセスの識別を示すオーナープロセス ID と、前記共有メモリ領域を参照するプロセスの数を示す参照プロセス数と、前記共有メモリ領域を参照中またはまだ参照していないプロセスの数を示す未参照プロセス数と、前記共有メモリ領域を参照するプロセスの識別を示す参照プロセス ID と、前記参照プロセス ID に示されたプロセスが前記共有メモリ領域を参照したのかどうかを示す参照済みフラグとを含むことを特徴としている。

【0017】前記共有メモリ空間に前記共有メモリ領域の使用状況を管理する共通管理情報部を備え、前記共通管理情報部は前記共有メモリ領域の 1 つ毎に対応した管理テーブルを複数持つことを特徴としている。

【0018】前記管理テーブルは、前記共有メモリ空間の前記共有メモリ領域の場所の識別を示す共有メモリ領域 ID と、前記共有メモリ領域の使用の有無を示す使用

中フラグと、前記共有メモリ領域を確保したプロセスの識別を示すオーナープロセスIDと、前記共有メモリ領域を獲得した日時を示す取得日時とを含むことを特徴としている。

【0019】

【発明の実施の形態】次に、本発明の実施の形態について図面を参照して説明する。

【0020】図3は、本発明の共有メモリを使用したプロセス間通信方式を適用したシステムの構成図である。

【0021】図3を参照すると、本発明の実施の形態は、共有メモリ空間1と、共有メモリ空間1内に共有メモリ領域（以降、シャトルバッファと呼ぶ）であるシャトルバッファ32と、プロセスA35と、プロセスB36と、プロセスC37と、プロセスD38と、シャトルバッファ32を使用するためにプロセスA35、B36、C37、およびD38が共有する共有メモリ空間1内にシャトルバッファを管理するための共通管理情報部31と、シャトルバッファ32の作成を行う作成手段2と、シャトルバッファ32の編集（参照、変更、または解放）を行う編集手段3を備えている。

【0022】共通管理情報部31には、複数の管理テーブル41が存在し、管理テーブル41は1つのシャトルバッファを管理するために1つ存在する。図3に示す通り共通管理情報部31の管理テーブル41のフォーマットとして、シャトルバッファIDと、使用中フラグと、オーナープロセスIDと、シャトルバッファの取得日時とが存在する。シャトルバッファIDは共有メモリ空間1に存在するシャトルバッファのアドレス位置に対応している。使用中フラグはシャトルバッファを使用しているか使用していないかを示し、「使用中」または「未使用」で表示される。オーナープロセスIDにはシャトルバッファを獲得したプロセスIDが設定される。取得日時にはシャトルバッファを獲得した日時が設定される。

【0023】システムの環境設定時に、システム全体で使用できるシャトルバッファの最大件数分、共有メモリ空間1に共通管理情報部31（管理テーブル41を最大件数分の領域）を用意する。

【0024】図3のシャトルバッファ32は、通信を行うとするプロセスA35が、共有メモリ空間1上を取得した領域であり、これは、制御情報部33、および通信データ部34から構成される。制御情報部33のフォーマットの詳細は以下の通りである。

【0025】aは、オーナープロセスIDを示し、シャトルバッファ32を取得したプロセスIDで、獲得したプロセスのみシャトルバッファ32の通信データ部34への書き込みが行える。

【0026】bは、参照プロセス数を示し、シャトルバッファ32を参照するプロセスの数を意味する。

【0027】cは、未参照プロセス数を示し、シャトルバッファ32を参照中、またはまだ参照していないプロ

セスの数を意味する。

【0028】dは、参照プロセスIDを示し、シャトルバッファ32を参照するプロセスのIDを意味する。

【0029】eは、参照済みフラグを示し、参照プロセスIDに表示されたプロセスについて、シャトルバッファ32を参照したのかどうかを示すフラグを意味している。。

【0030】図1は、図3の作成手段2の動作をフローチャートで示したもので、シャトルバッファ作成手順を示す。

【0031】図2は、図3の編集手段3の動作をフローチャートで示したもので、シャトルバッファの参照、変更、または解放手順を示す。

【0032】次に、図1、図2、および図3を参照して、本発明の実施の形態の動作について説明する。

【0033】まず、図1と図2を参照して、シャトルバッファの作成手順について説明する。

【0034】今、プロセスA35の処理を行うと、プロセスA35はプロセスB36、プロセスC37の処理をそれぞれ必要とし、プロセスB36の処理を行うと、プロセスB36はプロセスD38の処理を必要とする。

【0035】シャトルバッファを用いて通信を行おうとするプロセスA35は、共通管理情報部31から使用可能なシャトルバッファID（管理テーブル41の使用フラグが「未使用」であるもの）を検索する（図1のステップS11、S12）。シャトルバッファの空きがない場合は、空きがでるまで待たされる。

【0036】使用可能なシャトルバッファIDを見つけたら、共通管理情報部31の管理テーブル41において、シャトルバッファIDに対応する使用中フラグを「使用中」に、自プロセスIDをオーナープロセスIDに、システム日時を取得日時に設定する（ステップS13）。

【0037】次に、管理テーブル41のシャトルバッファIDから共有メモリ空間1上にシャトルバッファ32の領域を取得し、自プロセスIDをシャトルバッファのオーナープロセスIDに、通信先のプロセスID（プロセスB36、プロセスC37）を参照プロセスIDに、通信先のプロセス数を参照プロセス数に設定する。未参照プロセス数は、参照プロセス数と同じ値を設定する（最初は全て未参照なので、未参照プロセス数＝参照プロセス数＝2となる）。また、各参照プロセスの参照済みフラグは全て「未参照」に設定する（ステップS14）。

【0038】次に、プロセスA35はシャトルバッファ32の通信データ部34に通信データを設定し（ステップS15）、プロセスB36およびプロセスC37にソケット通信によりシャトルバッファIDを通知する（ステップS16）。

【0039】上記手順により、シャトルバッファが作成され、通信先プロセスにデータが受け渡される。

【0040】次に、図2と図3を参照して、シャトルバッファ参照、変更または解放手順について説明する。

【0041】前記手順により、プロセスA35からソケット通信によりシャトルバッファIDを通知されたプロセスB36およびプロセスC37は、通知されたシャトルバッファIDからシャトルバッファを参照する（図2のステップS21）。

【0042】プロセスB36は、プロセスD38にシャトルバッファ32の内容を受け渡すため、シャトルバッファ32の参照プロセス数および未参照プロセス数に1
10 加算し（未参照プロセス数=3）、プロセスD38のプロセスIDをシャトルバッファ32の参照プロセスIDに追加してその参照済みフラグを「未参照」に設定し、そしてプロセスD38にソケット通信でシャトルバッファIDを通知する（ステップS22、S23）。

【0043】次に、プロセスB36はシャトルバッファ32の自プロセスの参照済みフラグを「参照済み」に、未参照プロセス数を-1する（未参照プロセス=2）（ステップS24）。

【0044】ここで、シャトルバッファ32の未参照プロセス数をチェックするが（ステップS25）、未参照
20 プロセスは0でないので何もしない。

【0045】プロセスC37も同様にシャトルバッファを参照するが、他に通知するプロセスがないので、参照済みフラグの変更および未参照プロセス数のデクリメントのみ行う（未参照プロセス数=1）。ここでもシャトルバッファ32の未参照プロセス数をチェックするが、未参照プロセス数は0でないので何もしない。

【0046】プロセスB36から通知を受けたプロセスD38はシャトルバッファを参照後、参照済みフラグの
30 変更および未参照プロセス数のデクリメントを行う（未参照プロセス数=0）。ここで、未参照プロセス数が0になるので、シャトルバッファ32を解放して、共通管理情報部31のシャトルバッファIDの使用済みフラグを「未使用」に、オーナープロセスIDをクリア、取得日時をクリアする（ステップS26、S27）。

【0047】上記説明において、共通管理情報部31の管理テーブル41とシャトルバッファ32の状態遷移図を描くと図4の通りになる。

【0048】図4において、図4-（1）は、プロセスA（ID=0001とする）がシャトルバッファ（ID=5001とする）を取得（作成）し、プロセスB（ID=0002とする）とプロセスC（ID=0003とする）に通知するときの状態遷移図である。図4-
40 （2）は、プロセスB（ID=0002）が内容を参照後、プロセスD（ID=0004とする）に通知する状態遷移図である。図4-（3）は、プロセスC（ID=0003）とプロセスD（ID=0004）が内容を参照後シャトルバッファ（ID=5001）を解放した状態遷移図である。図4-（4）は、プロセスB、C、お

よびDの全ての参照が終わり、シャトルバッファ（ID=5001）を解放した状態遷移図である。

【0049】上記説明では、シャトルバッファIDを通知するためのプロセス間通信をソケット通信により行っているが、ソケット通信以外の手段を用いても可能なことは言うまでもない。また、図3ではプロセス数を4つにしてあるが、プロセス数がいくつであってもよいことはもちろんである。

【0050】以上説明したように、複数プロセスへの大量データのブロードキャスト通信、並びに受信データの他プロセスへの引き渡しを、高速、かつ容易に実現することができる。

【0051】また、全てのプロセスの参照が完了したシャトルバッファは、最後に参照を完了したプロセスが解放するので、プログラムの制御を簡単にすると共に、通信量の増大によるシステム性能の低下を防ぐことができる。

【0052】更に、多量の通信データをシャトルバッファを使用することにより、同じ領域（シャトルバッファ）で複数のプロセスに通知することができるため1対多数の通信ができる。

【0053】更に、通信用共有メモリ生成後の参照、譲渡、および解放の管理をシャトルバッファを使用することにより使用者に意識させない構造となるため、プロセス間通信の生産性が高まる。

【0054】更に、シャトルバッファと、共通管理情報部を設けることにより、管理情報を全て管理することとなり、状態遷移図でも示したように共有メモリの作成後の流れを容易にトレースすることができ、システム稼働後の不具合等の調査を容易に行えることができる。

【0055】なお、上記説明において、ソケット通信のソケットとは、OS参照モデルのTCP/IPネットワーク内のトランスポート層上でのプロセス間通信によるデータ送受信を行うアプリケーションプログラムを作成するためのAPI（アプリケーション・プログラミング・インタフェース）であり、ソケット通信とは、ソケットを利用した通信のことである。

【0056】

【発明の効果】本発明の共有メモリを使用したプロセス間通信方式は、前述の通り以下の効果をもたらす。

【0057】本発明の効果は、複数プロセスへの大量データのブロードキャスト通信、並びに受信データの他プロセスへの引き渡しを、高速、かつ容易に実現すること
50 できるということである。これにより、サーバから複数クライアントへの通信、並びに他プロセスへのデレゲーション（作業委託）通信を行うシステムを高性能かつ高生産で開発することできる。これは、本発明が、1つの共有メモリ領域を用いたプロセス間通信をメモリ管理情報を交換するための通信を行うことなく実現する方式を提供するからである。

【図面の簡単な説明】

【図 1】本発明の共有メモリを使用したプロセス間通信方式のシャトルバッファ作成手順を示すフローチャートである。

【図 2】本発明の共有メモリを使用したプロセス間通信方式のシャトルバッファ参照、変更または解放手順を示すフローチャートである。

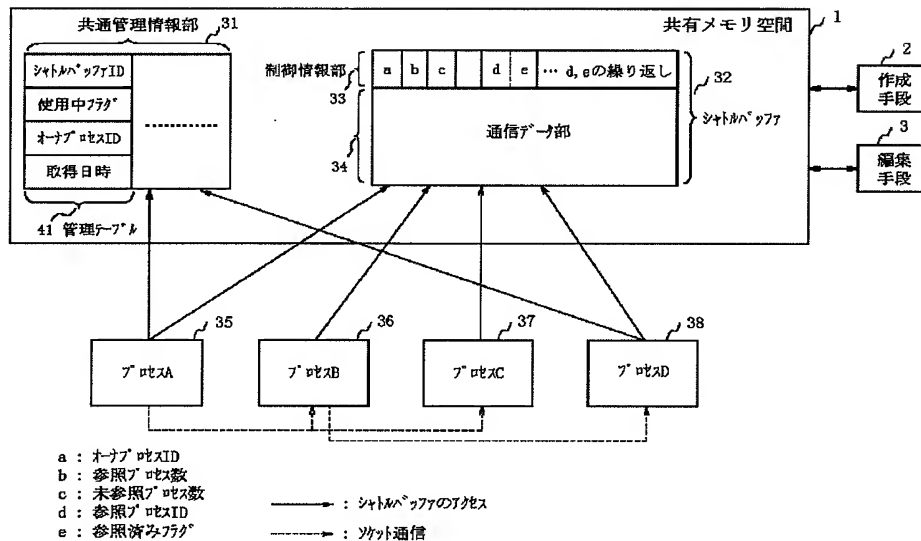
【図 3】本発明の共有メモリを使用したプロセス間通信方式を用いたシステムのブロック図である。

【図 4】本発明の共通管理情報部とシャトルバッファの関係を示した状態遷移図である。

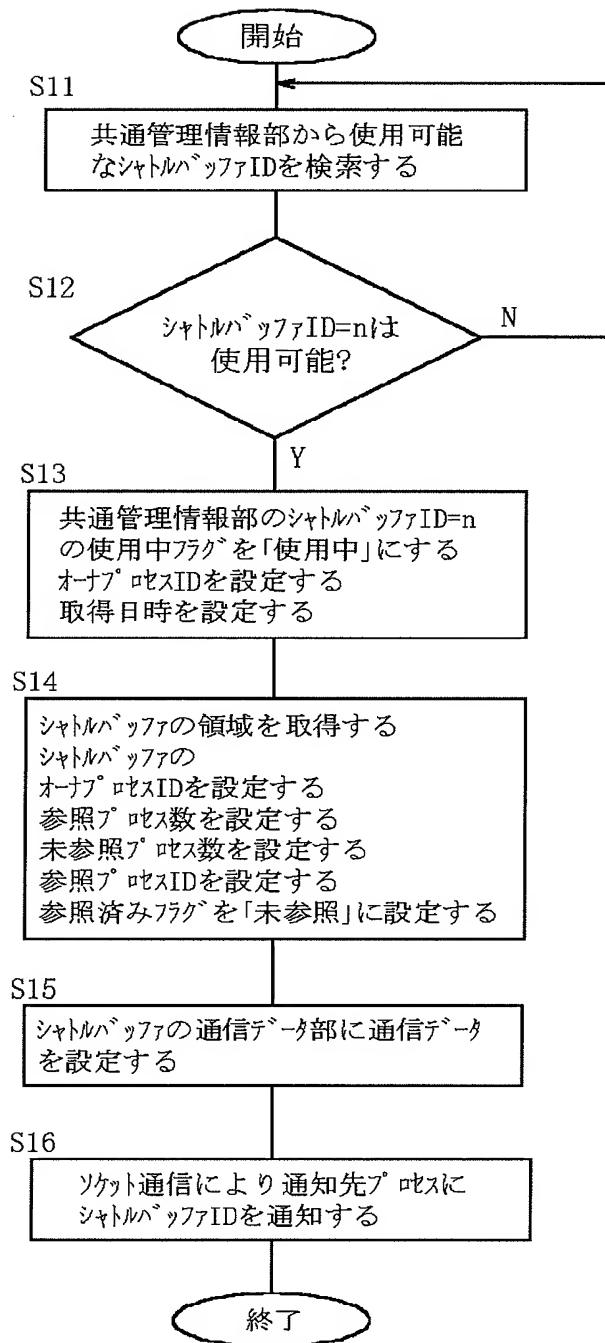
【符号の説明】

- 1 共有メモリ空間
- 2 作成手段
- 3 編集手段
- 3 1 共通管理情報部
- 3 2 シャトルバッファ
- 3 3 制御情報部
- 3 4 通信データ部
- 3 5 プロセス A
- 3 6 プロセス B
- 3 7 プロセス C
- 3 8 プロセス D
- 4 1 管理テーブル

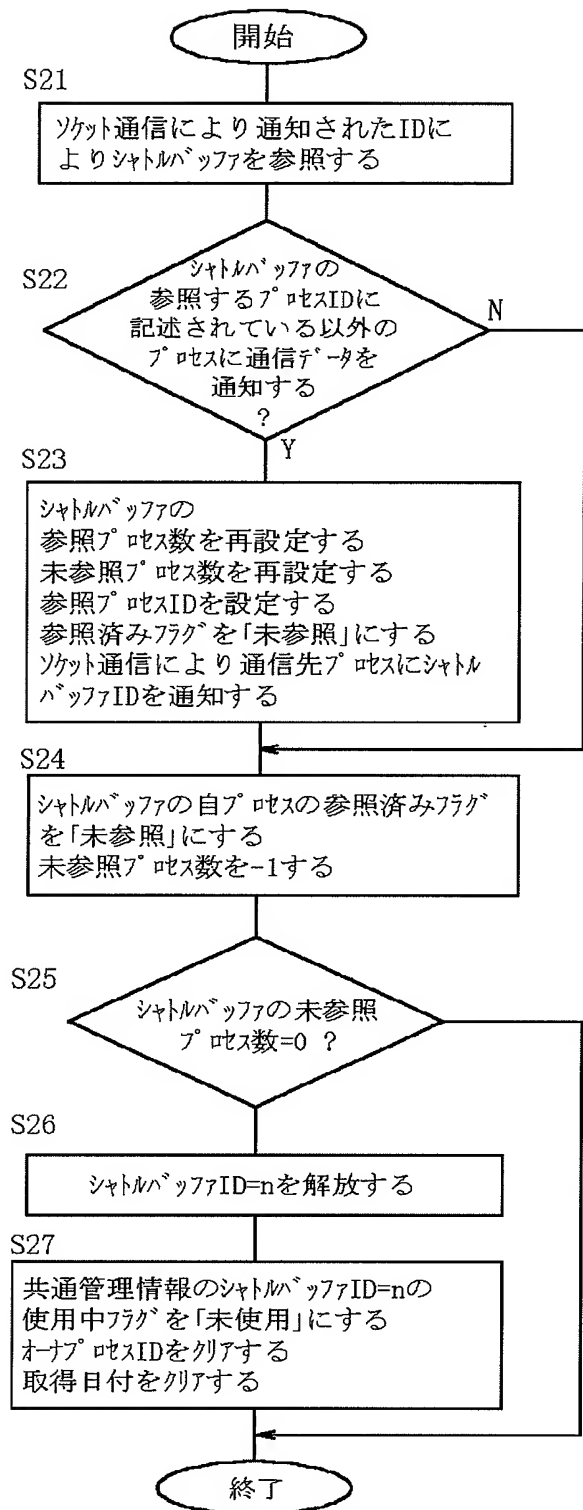
【図 3】



【図 1】



【図 2】



【図 4】

(1) 共通管理情報部31

5001	5002	
1	0	...
0001	0	
199703181200	NULL	

シャトルバッファID=5001

0001	2	2		0002	1	0003	1	0	0	0	...
通信データ											

↓

(2) 共通管理情報部31

5001	5002	
1	0	...
0001	0	
199703181200	NULL	

シャトルバッファID=5001

0001	3	2		0002	0	0003	1	0004	1	0	...
通信データ											

↓

(3) 共通管理情報部31

5001	5002	
1	0	...
0001	0	
199703181200	NULL	

シャトルバッファID=5001

0001	3	0		0002	0	0003	0	0004	0	0	...
通信データ											

↓

(4) 共通管理情報部31

5001	5002	
0	0	...
0	0	
NULL	NULL	

シャトルバッファが解放される